U.S. NAVAL ACADEMY
COMPUTER SCIENCE DEPARTMENT
TECHNICAL REPORT

Internet Protocol Version 6 (IPv6): Capabilities, Operational Concepts, and the Transition from IPv4

Domagalski, Joshua E.

USNA-CS-TR-2008-01

August 27, 2008

USNA Computer Science Dept. ◦ 572M Holloway Rd Stop 9F ◦ Annapolis, MD 21403

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
27 AUG 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Internet Protocol Version 6 (IPv6): Capabilities, Operational Concepts, and the Transition from IPv4	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **U.S. Naval Academy,Computer Science Department,572M Holloway Rd Stop 9F,Annapolis,MD,21403**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**32**	

U.S. NAVAL ACADEMY
COMPUTER SCIENCE DEPARTMENT
TECHNICAL REPORT

Internet Protocol Version 6 (IPv6): Capabilities, Operational Concepts, and the Transition from IPv4

Domagalski, Joshua E

December 6, 2007

Computer Science Department
SI495A: Research Project Report
Fall AY08

Internet Protocol Version 6 (IPv6): Capabilities, Operational Concepts, and the Transition from IPv4

by

Midshipman Joshua Domagalski, 081812

United States Naval Academy
Annapolis, MD

_____

_____
Date

Certification of Faculty Mentor's Approval

Assistant Professor Patrick Vincent
Department of Computer Science

_____

_____
Date

Assistant Professor Thomas Augustine
Department of Computer Science

_____

_____
Date

Department Chair Endorsement

Captain Thomas Logue
Chair, Department of Computer Science

_____

_____
Date

Executive Summary

The research study, *Internet Protocol Version 6 (IPv6): Capabilities, Operational Concepts, and the Transition from IPv4*, was conducted at the United States Naval Academy (USNA) with the aim of developing, employing and testing an IPv6 network while learning about many important compatibility and operational issues an organization would encounter while migrating to this new protocol. Specifically, the study entailed participation in the primary phase of the Defense Information Systems Agency's (DISA's) IPv6 Pilot Network Project. This primary phase required setting up a fully functional IPv6 network at USNA that, in turn, connected to an IPv6 network at the United States Military Academy (USMA) via an IPv4/IPv6 Virtual Private Network tunnel.

Although mandated by the Office of Management and Budget (OMB) to complete the critical conversion of IPv4 to IPv6 by 2008, the Department of Defense (DOD), recognizing the inherent security considerations, operational restrictions and resultant delays in conducting such a conversion, has sponsored the multi-phase US Service Academy IPv6 Pilot Network Project. By partaking in the pilot project, this research study facilitated the DOD-wide protocol conversion by making use of an IPv6 connection between USMA and USNA which afforded a secure testing and validation environment.

In addition to the inter-academy IPv6 network connection mandated by the Pilot Network Project, the scope of this research study was threefold: 1) to review and validate the prior research study conducted by the MIDN Christopher B. Landis, USN, 2) to test and develop convergence techniques for the coexistence of IPv4/IPv6, and 3) to discover and analyze the ramifications that the transition to IPv6 would have on legacy systems. It is noteworthy that during the course of the study, our findings revealed a disagreement with a finding reached in the previous Landis study (as might be expected when performing research on a new and largely unexplored topic). In addition, we found that it is vital to understand the IPv6 addressing scheme as it provides the critical and fundamental underpinning to the many other changes made to the new protocol.

By the conclusion of the study, we were able to create a fully functional IPv4/IPv6 network that connected to the USMA network via the tunnel. Although we attempted to create an IPv6-only network, we found that the use and integration of Microsoft's Windows XP PRO SP2 and Windows Server 2003 SP1 necessitated the existence of IPv4 on the network for the provision of some network services. In addition, we noted that the potential for transitioning from IPv4 to IPv6 using Linux-based operating systems seems much more promising than utilizing Microsoft's Windows XP.

Our research provides recommendations for future participation and study in this crucial DOD project.

1. Introduction

Security considerations, technical challenges, and operational requirements have hindered the Department of Defense (DoD) from investigating and testing Internet Protocol Version 6 (IPv6) on a large scale. In fact, DoD is seriously lagging behind current congressional and Office of Management and Budget (OMB) mandates to prepare for the transition from IPv4 (the protocol currently in use) to IPv6. In order to facilitate the conversion process, the Defense Information Systems Agency (DISA) sponsored a three-phase U.S. Service Academy IPv6 Pilot Network Project.

This research study was intended to serve as an integral part of phase one of the DISA Pilot Network Project. This research entailed building, operating and maintaining a pilot IPv6 network between the United States Naval Academy and the United States Military Academy. This network was then used to provide validation and refutation of operational concepts developed for the transition from IPv4 to IPv6, to include investigations of network management, address allocation and Domain Name Services. This research also investigated the testing and development of protocol capabilities and communication, as well as transition techniques for migrating from the IPv4 protocol to IPv6.

Although prevalently misunderstood as simply "IPv4 with a larger address space," IPv6 represents an entirely new protocol incorporating 4 major changes: 1) IP addresses are expanded from 4 bytes to 16 bytes, 2) the format of the packet header is simplified to include only seven fields (from 13 in IPv4) thus making routing faster, 3) various provisions are incorporated to enhance Quality of Service (QoS) and 4) security is improved through authentication and privacy capabilities. Notwithstanding these four major changes, the new protocol should be backwards compatible with IPv4. Current DoD networks, operating under IPv4, remain vulnerable to limited address space, antiquated architecture, a difficulty in providing quality of service for voice and video applications, and critical security concerns.

2. DoD Internet Protocol Version 6 Generic Test Plan (DoD IPv6 GTP) and DISA

In September 2006, in accordance with OMB mandates, the DoD established the goal of transitioning all DoD networks to Internet Protocol Version 6 (IPv6) by fiscal year 2008 [1]. Clearly, this target date will not be met. In order that this transition be eventually accomplished, DISA published the *Department of Defense Internet Protocol Version 6 Generic Test Plan* (GTP) which requires the testing, analyzing, and validating of commercial and government IPv6 implementation. Thus, in order for a product to be considered "IPv6 capable," it must complete testing for performance and interoperability in accordance with the Generic Test Plan (GTP) as well as be in conformance with the various industry-wide standards that the Internet Engineering Task Force establishes in its *Requests for Comments* (RFCs) [1]. This DoD IPv6 GTP "specifies test criteria and procedures for IPv6 products involved in or connecting to the Global Information Grid" (DISA) [1]. The GTP divides the testing of IPv6 into nine main categories: Core IP Functionality, Routing and Switching, Transition Mechanism, Common Network Applications, Security and Information Assurance, Mobility, Quality of Service, Multicasting, and Network Operations and Management.

As a subset to the overall DoD IPv6 GTP, the Defense Information Systems Agency sponsored the U.S. Service Academy IPv6 Pilot Network Project which aims at eventually connecting the five federal service academies—the United States Naval Academy (USNA), the United States Military Academy (USMA), the United States Air Force Academy (USAFA), the United States Merchant Marine Academy, and the United States Coast Guard Academy (USCGA)—together using an IPv6/IPv4 tunnel. This process of connecting the different academies was divided into phases: phase one, currently underway, entails the initial connection of USNA with USMA; phase two will involve the connection of USAFA to this network, and phase three will see the connection of USCGA and USMMA to this network.

The primary purpose of this Pilot Network Project was fourfold: 1) to "provide validation or refutation of operational concepts developed for transition from IPv4 to IPv6, to include investigation into Information Assurance, Network Management, Multicasting, Domain Name Services, and standard Transport Control Protocol (TCP) services," 2) to "develop common best practices for the utilization of IPv6," 3) to "develop IPv6 as a protocol by experimenting with its inherent capabilities for mobility, flexibility, robotics control, and convergence of services," and 4) to "provide training to the staff, faculty, and students in the next generation protocol to be used throughout the US Department of Defense" [2].

3. Addressing

The IPv6 protocol differs in many significant ways from IPv4; first and foremost in the addressing mechanism that is utilized. As addressing is a significant part of any form of communication, understanding the differences in IPv6 addressing involves not only the focus on the larger address space, but also the effects of addressing on routing algorithms as implemented in the next-generation protocol.

IPv4's 32-bit address offers, in theory, 4,294,967,296 unique addresses [3]. This seemed like a generous allowance of addresses when the Internet was first launched as a military and academic research project in the 1970's. With the growing popularity of the Internet beginning in the early 1990's, address exhaustion became a concern. Although Network Address Translation, Classless Inter-Domain Routing (CIDR), Dynamic Host Configuration Protocol and other short-term stopgaps have delayed the exhaustion of IPv4 address space, the fundamental problem remained unsolved while the solutions in some cases actually further complicated matters. Ergo, IPv6 provides a 128-bit long address, yielding 2^{128} (340,282,366,920,938,463,463,374,607,431,770,000,000) different unique IP addresses [4]. To shorten the notation used for the address and to simplify calculations, IPv6 addresses are represented in hexadecimal rather than decimal notation. This reduces the length of the address to 32 hexadecimal characters, divided by colons into 8 groups of 4 characters (16 bits) each. To further simplify the address notation, zeroes are handled in a special way. First, all leading zeroes are omitted from each group of four hexadecimal characters. Second, consecutive zeroes of any length can be omitted to further collapse the IPv6 address – two colons are used where this omission takes place. However, consecutive zeroes can only be omitted once to avoid ambiguity in reconstructing the full address. Thus, by following these notation rules, an IPv6, unicast global address, **2001:0000:0000:00A1:0000:0000:0000:1E2A**, can also be written as **2001:0:0:A1::1E2A** [4].

The IPv6 address architecture classifies addresses as one of three types: unicast, multicast, and anycast [5]. A unicast address is an address that identifies a single node. In other words, traffic sent to a unicast address will only be forwarded to a specific, single node. A multicast address, as the name implies, is an address that identifies a group of nodes (normally found on a given site). Traffic sent to a multicast address will be forwarded to each node within that group. Lastly, anycast is an address that identifies a group of nodes where any traffic forwarded to that anycast address will be forwarded to the nearest node within the group.

Unicast addresses can be further divided into three types: link-local, site-local (sometimes referred to as "unique local"), and global [4]. All unicast addresses contain both a network prefix and an interface identifier: the network prefix denotes the link while the interface identifier denotes the exact node. Link-local addresses identify hosts on a single link (layer 2 domain). This address is normally used for neighbor discovery, automatic address configuration, or when routers are not present on the network. Link-local addresses are assigned the address space **FE80::/10** (where the familiar CIDR slash notation is carried over from IPv4, in this case indicating that all 128-bit link-local addresses will always begin with the 10 bits: 1111111010) [4]. Site-local addresses identify hosts on a single domain or site using the network prefix. Normally, a site-local scheme will be pushed on the site via a router. These addresses are also referred to as unique-local addresses (ULAs). Site-local addresses will be found in either the **FC00::/7** or **FD00::/8** address space (this is normally locally assigned). For both link-local and site-local addresses, routers will not forward any packets with these addresses as source or destination outside of the given site (similar to private addressing schemes in IPv4). In addition, the 8^{th} bit of the address prefix facilitates the possible, future ability to identify an assignment policy [4]. Global unicast addresses (GUAs), however, are forwarded outside of the site or domain by routers as their purpose is to provide a unique unicast address available globally. These addresses can be identified by their high-level 3 bits being set to 001 (**2000::/3**) [6].

Of particular interest in the design of unicast addresses is the integration of the "interface identifier" which is unique for each host [4]. The interface identifier occupies the lower 64-bits of all unicast addresses and is related to the Media Access Control (MAC) address. In order to generate an IPv6 Interface Identifier from a MAC Address, several steps must occur. First, the 48-bit MAC address is taken and divided exactly in half. These two halves are then buffered with 16-bits (**FFFE** is inserted in between the two halves). The result is the EUI-64 (Extended Unique Identifier) representation [4]. After the EUI-64 identifier is obtained, the seventh bit of the 16 high-level bits is "flipped". The result is the IPv6 Interface Identifier. The following provides an example of how this is done for one of the computers (named PANDORA) used in USNA's lab:

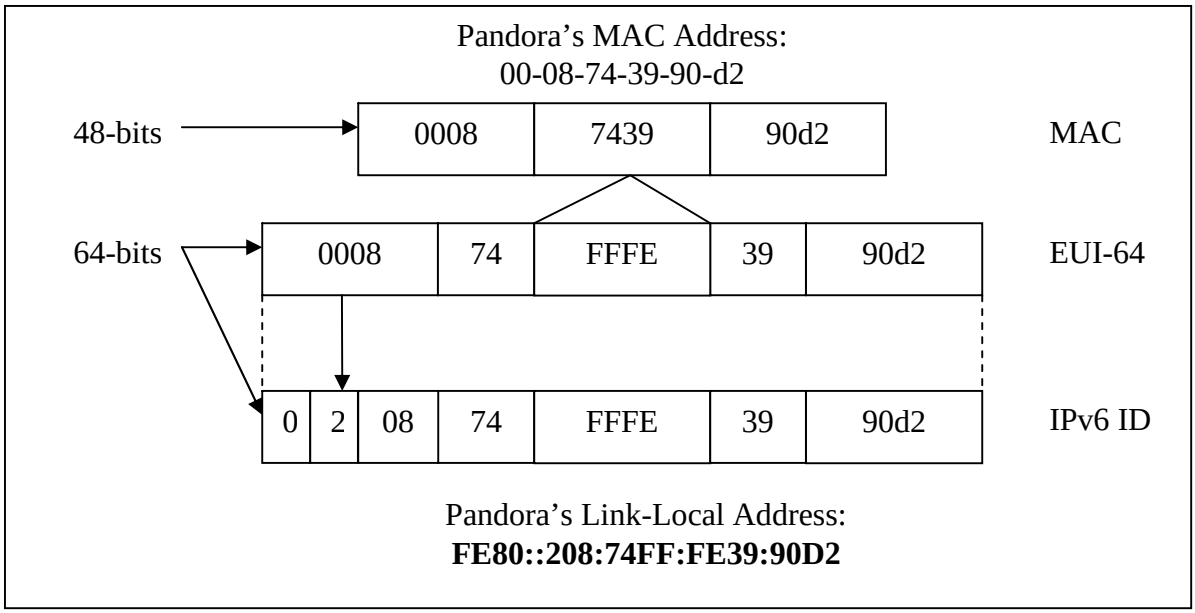

Figure 1

Because link-local and site-local addresses can, by their very nature, be reused, RFC 4007 employs the use of a "zone identifier" (also known as a "scope identifier") [7]. As an example, PANDORA's link-local address would be seen as the following: **FE80::208:74FF:FE39:90D2%5**. Here, the IPv6 prefix and interface identifiers are the same; but, in order to distinguish the scope, the %5 indicates the "zone". The main purpose of having zone identifiers is to make a link-local address unique for a node that is connected to multiple links. However, this syntax presents a conflict with RFC 2396 in regards to Uniform Resource Identifier (URI) Generic Syntax [8]. Common examples of this conflict in syntax can be seen in the following example of a simple search query:

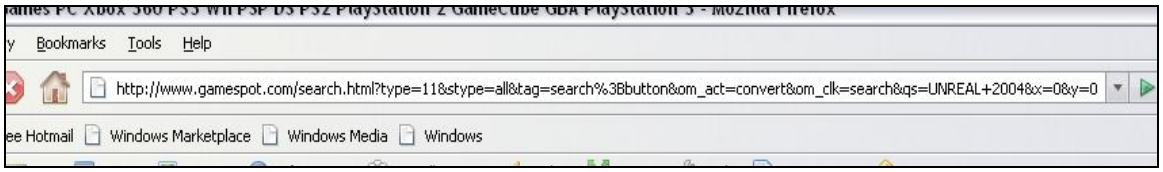

Figure 2

According to RFC 2396, '%' is reserved for the special purpose of encoding escape characters [8]. While generally not presenting difficulties as most METHOD="GET" separates the URL from the appended search query forming the URI, the potential for conflict is not beyond the realm of possibility.

In addition to the prefix and interface identifier segments of unicast addresses, Global and Subnet ID segments are also provided. Though originally meant to be implemented in a global aggregation scheme, the Global and Subnet IDs are now controlled by the Internet Assigned Numbers Authority (IANA) and the respective service provider (SP). The global prefix is 48 bits or shorter and is allocated to a service provider by IANA [9]. The subnet ID identifies the organizational structure of its network in respect to the service provider.

Another important change in the IPv6 protocol is the replacement of broadcast addresses with multicast addresses. Basically, a multicast address identifies a group of

interfaces so that a packet with a multicast destination address is sent to all those belonging to the multicast group.

All IPv6 multicast addresses have their first 8 (high-order) bits set to 1 thus giving the address structure **FF00::/8** [4]. After the first 8 bits, the next 4 bits act as a flag of which the three lower-order bits are currently used. The second flag bit, 10^{th} bit from the beginning, indicates whether that multicast group address contains the unicast address of the RP (rendezvous point) [4]. The third flag bit, 11^{th} bit from the beginning, indicates whether the multicast address is based on a unicast prefix or not. The fourth flag bit, 12^{th} bit from the beginning, indicates whether it is permanently assigned or not. It is important to note, however, that if the third flag bit is set to 1 indicating that it is based on a unicast address, then by necessity the fourth flag bit must also be set to 1 indicating a nonpermanent address. This is due to the fact that because a unicast address has a limited lifetime, and the multicast address in this case is derived from the unicast address, the multicast address must also have a limited address [4].

The last change in IPv6 regarding address types is the anycast address. According to RFC 3513, an anycast address is a unicast address assigned to multiple machines and is routed to the nearest interface configured for it [5]. However, as anycast addresses are virtually indistinguishable from unicast addresses, a given node must be configured for the assignment of an anycast address to its interface. Because an anycast address is assigned to multiple machines, it cannot be used as the source address for a packet. Oftentimes, the anycast address is used in the replication of important network resources such as web servers, multicast RPs, and DNSs which can allow for the sharing of traffic loads [4].

In summary, IPv6 provides a new address scheme that was meant to address many of the problems and shortcomings of the IPv4 addressing scheme. Although the change in the addressing scheme is not necessarily the biggest change in IPv6, it is probably the most noticeable and necessarily created numerous implications on the research study that was conducted.

4. Research Study

Primary in purpose to this research study was the establishment of a basic IPv6 network; therefore, fundamental to its inception were the use of basic and commonly used software and operating systems. Rather than immediately attempting to connect to the United States Military Academy or rush headlong into the implementation of a router under IPv6, we instead set up a simple three computer and one hub configuration as the most pragmatic foundation upon which to build. The OS chosen for this initial configuration was Windows XP SP2.

A. IPv6 Compatibility with Windows XP SP2

Before beginning the installation and setup of an IPv6 network, IPv6 compatibility with Windows XP SP2 had to be confirmed. After some research, it was discovered that IPv6 is preinstalled as a package—though not set up—on Windows XP SP2 platforms. This is in stark contrast to Solaris 10, SUSE 10.1 and other Linux flavors which, by default, have IPv6 enabled.

A user can setup up IPv6 on their machine, creating link-local addresses as described above, by using the "netsh" command-line utulity. The netsh tool also provides the ability to configure a router on a Windows XP system with forwarding and advertising enabled, to create static routes, and to assign site-local IPv6 addresses.

To enable IPv6 on each computer, we used the command:

netsh interface ipv6 install

Then, for the machine named PANDORA, we used the command to initiate the sending of "Hello" packets:

netsh interface ipv6 set interface "Local Area Connection" forwarding=enabled advertise=enabled

This command caused PANDORA, while running Windows XP SP2, to advertise its address on the interface named "Local Area Connection."

At this point, our three machines (PANDORA, DAEDALUS and ICARUS), had the following link-local addresses auto-configured:

PANDORA: **FE80::208:74FF:FE39:90D2**

DAEDALUS: **FE80::208:74FF:FE39:90D4**

ICARUS: **FE80::208:74FF:FE39:9105**

Note: At this point, the zone identifier (discussed in Section 3 above) was observed as "%5" and was attached to the end of the link-local addresses.

The link-local addresses also empirically confirmed that IPv6 link-local addresses incorporate the MAC address of the machine into the Interface Identifier which comprises the last 64 bits of the link-local address. In an effort to determine whether or not the MAC address implementation for IPv6 was at the Data Link Layer (MAC sublayer) or whether the implementation was higher, MAC spoofing was conducted using T&R SoftNet Solutions' MACSpoof to spoof the MAC address. The IPv6 link-local address change was successful. This proved that IPv6 uses the EUI-64 identifier derived from the 48-bit IEEE 802 address (RFC 2464) and that the implementation of the MAC address was on a higher layer than the Data Link Layer.

B. Basic Connectivity

After installing the IPv6 package on each of the three computers, they were connected via a 4-port hub as shown in Fig. 3. Pings (i.e., ICMP echo request/echo reply exchanges) were successfully sent between all of the computers using the IPv6 addresses.

The IPv6 literature (such as it is) discusses the use of a command named ping6. We learned that "ping" is IPv6 compatible simply by context, and "ping6" is not needed. This is because the initial difference between ping and ping 6 was due to the syntactical

difference between IPv4 and IPv6 addresses. However, as of Service Pack 2, Windows XP allows for the use of the normal "ping" command followed by the IPv6 address.

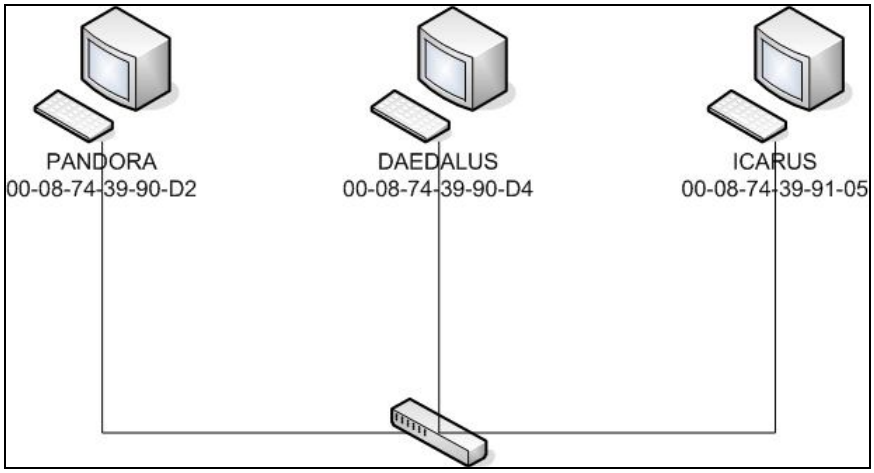

Figure 3

But was this really IPv6? In order to test that the actual packets being sent consisted of IPv6 traffic, Wireshark was installed on the three computers. Wireshark is a protocol analyzer that allows the user to examine in detail the raw traffic being placed on a network. Additionally, Wireshark incorporates the filter "ipv6" to readily filter IPv6 traffic. Using Wireshark, we confirmed that IPv6 was indeed being used for neighbor advertisements, neighbor solicitations, and pings; this traffic was also marked as "ICMPv6" in the Wireshark display.

To discover the network's Maximum Transfer Unit (MTU), i.e., the maximum-length packet that could be sent without fragmentation, we pinged ICARUS using DAEDALUS, increasing the buffer size each time until reaching fragmentation. The result was 1452. Since the ICMPv6 header contains 6 bytes, and the IPv6 header contains 40 bytes, the maximum buffer size is 1500 bytes (1452 + 8 + 40), as expected, since Ethernet is the underlying link layer protocol.

On a side note, we came to the lab and noticed all the machines were powered off. Upon powering on the machines and checking the physical setup, we began to ping the machines using their link-local addresses with the zone identifier that was attached on the preceding day. The result was "Destination Unreachable". Upon further investigation, we used the command:

netsh interface ipv6 show interface

which outputs the local index for each given link-local IPv6 address. We learned that this zone identifier can be different each time the machine is powered-up as it is dynamically determined by the node on the link. If this zone identifier is not used in the ping command, or if an incorrect one is used, the user will receive an error "Destination Unreachable."

C. Telnet and FTP

After establishing the initial setup, we decided to test basic services so that data communication and transfer (other than ICMPv6) could be empirically observed and validated.

We first tested telnet terminal emulation services. After initial user/login setups and the opening of port 23 in the Windows Firewall, a successful telnet connection was achieved between DAEDALUS (**FE80::208:74FF:FE39:90D4**) and ICARUS (**FE80::208:74FF:FE39:9105**) using standard telnet commands. The connection was confirmed and referenced with IPv6 addresses using the netstat command.

Our next goal was to test FTP (File Transport Protocol). This was initially unsuccessful. Upon investigation, we first considered that it must have been due to the Windows Firewall. After inspecting the settings and insuring that the correct port was enabled, we attempted setting up the FTP using a different port that would also be allowed by Windows Firewall. Unfortunately, the results were still negative.

Due to the ensuing problems we faced with as basic a protocol as FTP, we researched Microsoft's exciting whitepapers. However, we immediately saw a conflict in Microsoft's own documentation. As can be seen below in Fig.4, one of Microsoft's websites states that FTP is IPv6 supported.

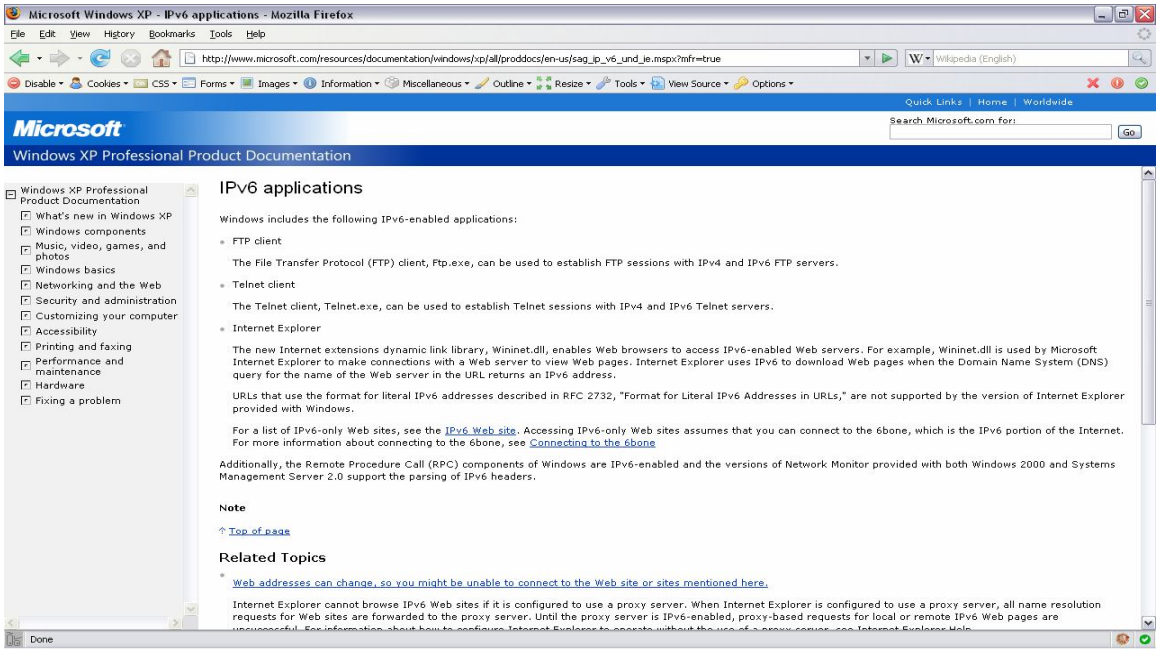

Figure 4

However, another Microsoft website, as seen below in Fig 5, notes that Microsoft's Internet Information Services (IIS) version 6.0 is not fully IPv6 compatible. IIS provides Internet-based services for FTP, Simple Mail Transfer Protocol (SMTP), Network News Transfer Protocol (NNTP) and web. As can be seen from Fig 5, under IPv6 FTP, SMTP and NNTP are not supported.

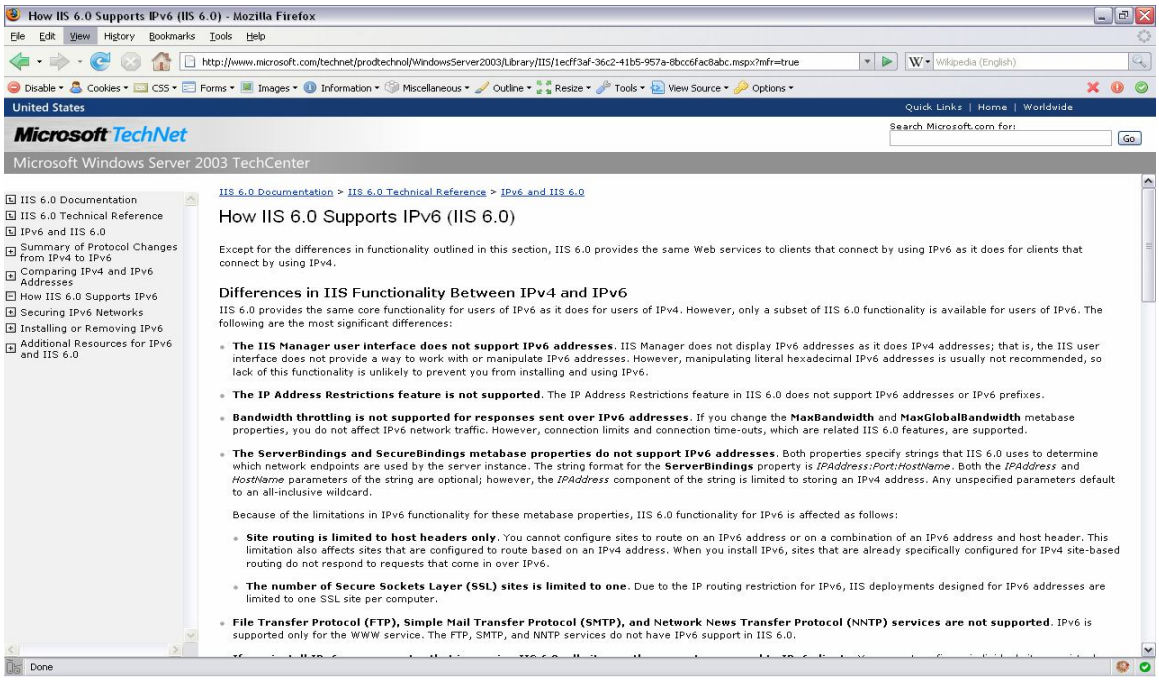

Figure 5

This created some confusion. Our test results revealed agreement with the statement in Figure 5, but disagreed with Figure 4. Another point of interest is the fact that the site that states FTP support under IPv6 is from the Windows XP Professional Product Documentation (the documentation for the operating system being used) as opposed to the Microsoft Windows Server 2003 TechCenter site (Fig 5). Still yet another point of conflict is the fact that Windows XP PRO SP2 utilizes IIS 6.0 for many of its internet and networking services. We finally determined that Microsoft's TechNet information (Fig 5) was the most authoritative word on the matter of FTP/IPv6 compatibility. In addition, the EnableReverseDnsLookup property is not supported, thus not allowing for lookups determining the DNS name of a client computer. This is important because EnableReverseDnsLookup is used by IIS 6.0 to perform reverse DNS lookups for the name of the client computer. This is the primary reason FTP does not work natively on Windows XP using IPv6.

Having determined that Microsoft's innate FTP service was not IPv6 compliant, it remained to be determined: Is there an IPv6 FTP client/server program available anywhere?

To determine whether FTP was feasible with IPv6, free, third-party software was installed and utilized (Xlight FTP). We loaded the XLight FTP virtual server on PANDORA. We then selected an option that allows "**Dynamic IPv6**" on well-known **port 21**, setting up user a username and password, we then created a remote admin account on **Port 3333.**

FTP connection was successful from the command line; however, there were initial problematic issues. Upon attempting to initiate a file transfer using the command:

ftp fe80::208:74ff:fe39:90d4%5

we could not execute the DIR command (we would receive "list filed/directory: FAILED") and the FTP command line would state the following and then freeze:

150 Opening ASCII mode data connection for /bin/ls (x bytes)

where x was the number of bytes.

Fortunately, these issues were resolved after restarting the machines after installing XLight FTP. Once these initial problems were resolved, several files of differing lengths were sent, all with 100% success. This demonstrated that FTP is possible with IPv6, but that it is not natively compatible with Windows XP.

D. Expanding the IPv6 Network

After having tested the telnet and FTP on the Windows XP machines, we installed Sun Solaris 10 on three other machines and then connected them to a different hub. Of interest, there was no need to install or setup any IPv6 support on the Sun Solaris 10 machines as IPv6 is, by default, already setup. This hub was then connected to the other hub creating the network layout as shown in Fig 6.

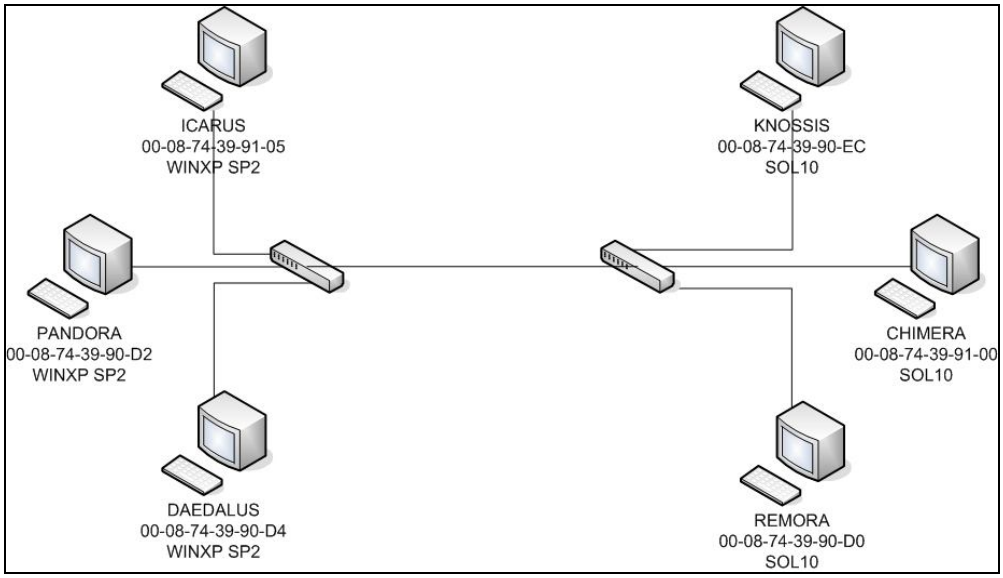

Figure 6

At this point, the lab was set up with the following link-local addresses:

- PANDORA -> **FE80::208:74FF:FE39:90D2**
- DAEDALUS -> **FE80::208:74FF:FE39:90D4**
- ICARUS -> **FE80::208:74FF:FE39:9105**
- REMORA -> **FE80::208:74FF:FE39:90D0**
- CHIMERA -> **FE80::208:74FF:FE39:9100**
- KNOSSIS -> **FE80::208:74FF:FE39:90EC**

After connecting the hubs, we began to successfully ping the Sun machines from the Windows machines. We then were able to successfully telnet between the Sun and

Windows machines. In addition, File Transport Protocol is natively supported by Sun Solaris 10 when using IPv6.

As we now had some of the basic services running for the network, we decided to install Windows Server 2003 SP1 Enterprise Edition on PANDORA in order to test FTP, DNS, DHCP, and Web Server services. Thus the network was designed as seen in Fig 7.

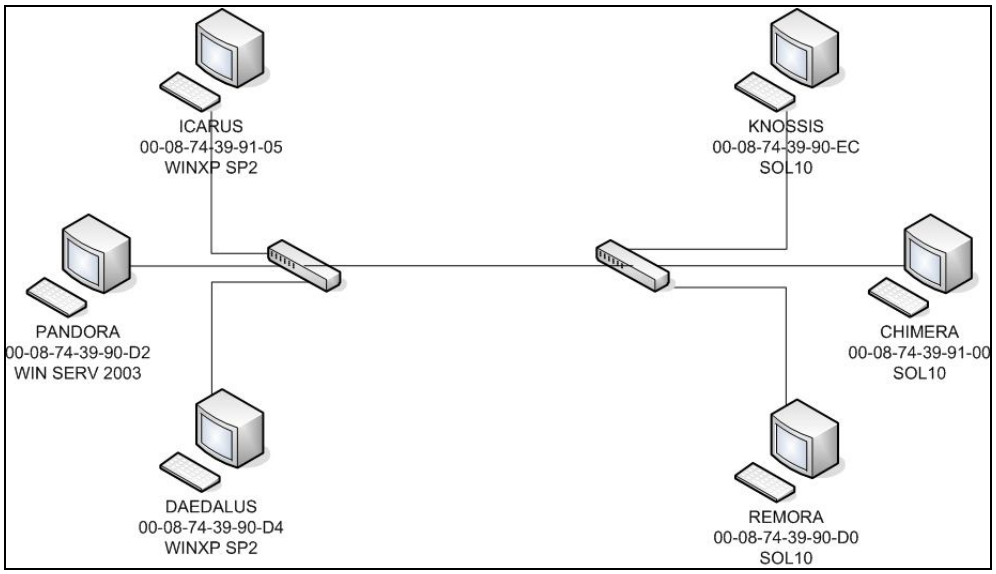

Figure 7

Initially, we set up a File Server on PANDORA (which now has Windows Server); but due to the same problem of IIS 6.0 incompatibility with IPv6, the File Server was not functional using IPv6. For example, upon entering:

ftp fe80::208:74ff:fe39:90d2%5

we received the error message

ftp: connect :unknown error number

At this point, our observation simply concurred with our previous conclusion about the incompatibility of IPv6 with IIS 6.0.

E. Establishing a Domain Name Service (DNS) Server

After testing and determing the incompatibility of IPv6 with the File Server on PANDORA, we determined to set up a Domain Name Service (DNS) server. However, due to the complexity involved, we decided to use Microsoft's *Step-by-Step Guide for Setting Up IPv6 in a Test Lab*[10]. This publication initially suggests setting up a network with three segments utilizing two systems as routers. Because the machines in the lab had only one Ethernet port and one Network Interface Card (NIC), and in order to simplify the setup, only one segment was created. This configuration was also chosen as

it most closely represented our current setup with the two hubs and six machines as shown in Fig 7.

We quickly learned something very important: Microsoft's guide does not create an IPv6-only network, but rather uses IPv4 for the DNS setup. Later, in fact, it was determined that Microsoft's DNS on Windows Server 2003 does not support DNS solely using IPv6.

Realizing that Microsoft was not quite ready for IPv6, we nevertheless, after following the guide, created a network with the following setup:

- IP: **10.0.1.0/24**
- IPv6 (site-local): **2001:DB8:0:1::/64**
  - Note: This was our first use of a unicast address that was not link-local.
- Each machine had the following IPv4 private addresses:
  - PANDORA -> **10.0.1.2/24**
  - DAEDALUS -> **10.0.1.3/24**
  - ICARUS -> **10.0.1.4/24**
  - REMORA -> **10.0.1.5/24**
  - Chimera -> **10.0.1.6/24**
  - KNOSSIS -> **10.0.1.7/24**
- Each machine had their Gateway set to **10.0.1.1**

After setting up the individual IP addresses, the following forward lookup zone was created: **testlab.ipv6.com**. This was possible because there was no DNS forwarding, and as there was no connection to the Internet, the use of this .com space is allowable. The DNS address was set up as **FEC0:0:0:FFFF::1%1**. This address was set as the DNS address for each system as it is one of three default IPv6 DNS addresses. For each computer the DNS suffix needed to be appended using the following commands:

- **Start->"Control Panel"->"Network and Internet Connections"->"Network Connections"**
- **Right-click "Local Area Connection"->Properties->General->"Internet Protocol (TCP/IP)"->Properties->Advanced->DNS**
- Add **testlab.ipv6.com** to the list of DNS suffixes.

In addition to appending the DNS suffixes, a static route was created in the following manner:

- **netsh interface ipv6 add route 2001:db8:0:1::/64 "Local Area Connection" publish=yes**

This command was necessary to create a static route and a site-local addressing scheme of **2001:db8:0:1::/64**. As there were no other routers on the test network, there was no need to add a next hop to the static route. However, in the case of using one DNS over a tunnel or using multiple VLANs, one would most likely setup a static route with multiple hops.

This setup provided the network with the site-local address scheme of **2001:DB8:0:1::/64**. It is of importance to note that until this point, the use of link-local addresses is more than sufficient for the maintenance of a network. However, after implementing certain network services often considered crucial to modern day networks, the assignment of a site-local addressing scheme to the link is necessary. As was the situation with the link-local addresses, the EUI-64-based interface ID was again appended onto the end of the site-local address scheme giving, for example, PANDORA's IPv6 address as **2001:0DB8:0000:0001:0203:74FF:FE39:90EC**. After the network was setup, AAAA DNS records (IPv6 DNS records) were created for each host (the site-local address was used).

As Sun Solaris 10 lacks much of a GUI for the assignment of gateways and a DNS, several files needed to be modified in the Sun Solaris 10 etc/ folder as follows:

- Modified **/etc/nsswitch.conf**
  - It was modified for the IPv4 and IPv6 address being used in the setup.
- Created **/etc/resolv.conf:**
  - **search testlab.ipv6.com**
  - **domain testlab.ipv6.com**
  - **nameserver 10.0.1.2**
- Modified /etc/inet/netmasks
  - Changed IPv4 address scheme to 10.0.1.0 255.255.255.0
- **ifconfig elxl0 10.0.1.**x **netmask 255.255.255.0 up** [where x denotes what machine this is being configured on]

These commands allow for the configuration of the Sun Solaris 10 computers to use the Windows Server 2003 DNS and lookup names for IPv6 addresses.

After setting up the DNS and creating the IPv6 DNS records for each host, we tested by pinging from each computer using the computer name rather than the computer's site-local address. After the testing was successful, we decided to test a telnet session using the DNS host name which was again successful. However, nslookups were not feasible as a reverse lookup zone was not created. After creating a reverse lookup zone and inserting the DNS host name records for IPv6, nslookups on client computers were successful.

Because DNS was functional and had tested successfully with IPv6, we decided to set up a Web Server on PANDORA. We created a webpage ("index.htm") and made it available to the public in the permissions. After setting up the web server, we attempted to connect to the website using Mozilla Firefox. Firefox successfully connected to the web server if we followed the following format:

http://[2001:DB8:0:1:208:74FF:FE39:90D2]/index.htm

The reason for the brackets around the IPv6 address is to prevent a parsing of the IPv6 address and the misinterpretation of the colons as the port to which to connect. Firefox also successfully connected using DNS names and bypassing the IPv6 address with the following format: http://PANDORA/index.htm. Internet Explorer v6.0, however, is not

IPv6 compatible. We attempted both the use of the IPv6 address in brackets as well as the use of the DNS name – both were unsuccessful.

F. Extending the IPv6 Network to the US Military Academy

Prior to connecting the network of computers to the United States Military Academy, we attempted to establish a clear IPv4 connection via the already established tunnel. An initial ping of 134.122.18.81 (the IPv4 address of USMA's router) resulted in a 0% success rate as each ping timed out. Running a trace route, we found the following results:

```
    Tracing route to 134.240.18.81 over a maximum of 30 hops

      1    < 1 ms       <1 ms          <1 ms         gw.ialab.usna.edu [131.122.204.254]
      2   188 ms       164 ms         155 ms         usna-rt9.swat.usna.edu
[192.190.228.1]
      3   162 ms       142 ms         160 ms         10.10.1.51
      4   159 ms       147 ms         162 ms         sdp1.usma.dren.net [138.18.9.49]
      5     *          278 ms         245 ms         cperouter.usma.dren.net [138.18.44.2]
      6     *          *              *              Request timed out.

    Trace complete.
```

Figure 8

This represented either a firewall issue, a non-existent router, or an IP address change. As USMA had been undergoing an address migration, the possibility for a possible IP address conflict was possible. After contacting Mr. Erik Dean at USMA, it was determined that the lack of "reachability" was due to the router's access control list (ACL) not responding to ICMPv4 packets.

After discovering the inability to ping using ICMPv4 packets, we decided to test a connection using USMA's IPv6 address (**2001:1918:F100:1::1**). This success rate, however, was 46% successful. After several re-attempts, the best success rate we could achieve was 60%. What we noticed was that every other packet was dropped by USMA. After contacting USMA, Mr. Dean replied with the following:

```
"I did some poking around in our router, we had our IPv6 address
assigned to an actual interface (one IPv6 address per interface)
instead of using the Loopback 6 address as COL Beckman originally
configured. This was leading to substantially less roundtrip ping
times between us and you. I reconfigured our router so that the
interface that you are connected to (via the tunnels) is now 'ipv6
unnumbered' and uses our 'Loopback 6' interface (which now has the
addresses 2001:1918:f100:1::1/64 and fec0:1918:f100:1::1/64). If
you ping the fec0 address you should see a very fast turnaround
time, the 2001 address may take a bit longer as the traffic has to
be routed, internally, by our router and then sent back."
```

Figure 9

After USMA reconfigured their router, we achieved 100% success rates when pinging USMA with their IPv6 address. At this point, we connected the network we had built to the CISCO 3600 router. After deleting the route **2001:DB8:0:1::/64**, the network took

the site-local scheme of **2001:1918:F101::/48**, as per the setup of the 3600 router (it was given pre-configured by ITSD). After a short delay, all of the machines were "pingable" using the new site-local scheme, the web server still worked by the new site-local address, and telnet and FTP (under the conditions as mentioned above) worked. At this juncture, we had achieved the goal of the first phase of the DISA Pilot Network Project. Our network configuration can be seen in Figure 6.

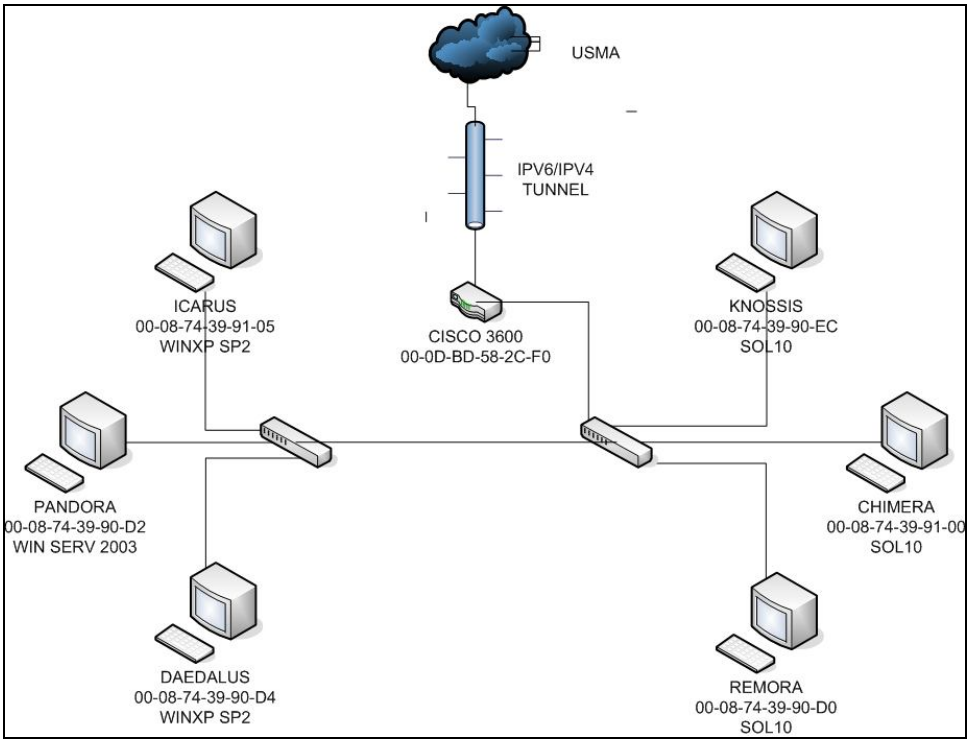

Figure 10

After we had achieved the goal of a network functioning via an IPv4/IPv6 tunnel with USMA, we decided to setup the DNS as strictly IPv6. However, this attempt failed, most likely due to static routing issues. This necessitated the creation of static routes for the CISCO 3600 router.

Towards the end of the research study, we desired to be able to test the bandwidth and data transfer rates and times. This necessitated the ability to synchronize the clocks on the computers. In order to do this, we attempted to set up a Network Time Protocol (NTP) server on the network by which all machines would sync their clocks. However, the NTP service as provided by Windows is not IPv6 supported. Furthermore, Windows necessitates an Active Directory for the setup and maintenance of an NTP service. Because we were unable to synchronize the clocks on the machines, we were unable to test the data transfer and throughput of the network.

5. Results

In this research study, we built a fully functioning network that implemented IPv6. We were also able to test and validate the use of common, mainstream operating

systems such as Windows XP Professional SP2 and Sun Solaris 10. Through the creation of an IPv4/IPv6 network, we were able to implement a DNS service that worked for both forward and reverse lookups; allowing nslookups, pings, file transfers, and telnet services based on names rather than IPv6 addresses. After the creation of the network and the establishment of common network services, we were able to connect the network to an IPv4/IPv6 VPN tunnel; thus completing the primary phase of DISA's Pilot Network Project. In addition, we were able to validate the address changes of IPv6 and were successful in the manipulation of those addresses to different address schemes.

We determined that FTP as implemented by Microsoft's IIS 6.0 is not IPv6 compatible. As this is predicated upon the incompatibility of EnableReverseDnsLookup with IPv6, many other services supported by IIS 6.0 are also incompatible: DHCP, Active Directory, SNMP, and NTP. However, DNS is compatible if the network is an IPv4/IPv6 network. In addition, our findings indicate that for the web server, Internet Explorer v6.0 is also not compatible with IPv6.

6. Recommendations

In order to better facilitate the further testing of IPv6, especially in regard to QoS, DNS setups over VPN tunnels, file streaming, and throughput testing, it would be beneficial to conduct a research study inclusive of students at both Academies working together. This would allow for a more realistic interpretation of results, allocate work more effectively, and allow for the sharing of critical resources that one or the other location may lack.

Furthermore, another recommended program enhancement would be the acquisition of network analyzing programs. An example of this type of program is Cisco's NetFlow Analyzer, which is fundamental to Cisco's testing of throughput and other network metrics. This could allow for a more effective measurement of the network that would not be dependent solely upon an NTP service or free, third-party software.

Much within the realm of possible avenues for researching IPv6 is limited by the nature of the VPN connection with USMA on a two-port router. This fundamentally limited the ability to test connections with the actual nascent IPv6 Internet, the use of IPv6 NTPs on the Internet, and the testing of protocols more specific to the use of the Internet with regard to IPv6. If a router could be acquired with greater than two ports and implemented, it would then be possible to study an actual application of IPv6 to the IPv6 Internet.

7. Conclusion

The goals we established from the onset of the research study were three fold: 1) review and validation of the previous "lessons learned" from the prior Landis study and various other research findings, 2) test and develop IPv4/IPv6 convergence techniques involving a fully-functioning IPv4 network, and 3) test and develop inter-protocol communication and transition techniques specifically including legacy systems. In addition, it was an underlying goal to partake in and accomplish the initial phase of DISA's Pilot Network Project.

Initially the conclusions of this study disagree with the findings in the Landis study which stated that "IPv6 is not a new protocol" [11]. Though it is perhaps true that the lengthy time for the implementation of the conversion project is more due to the fact that the creators and implementers want to solve problems associated with established architecture and not create new problems associated with new protocols, a cursory look at something as superficial as the addressing structure demonstrates a fundamental change in the protocol; something that indeed reflects something "new". However, it is probably more credible that the reason for the slow conversion to IPv6 is more due to the wide standardization, implementation, and acceptance of the known IPv4 protocol. Obviously, this makes it difficult to implement the new IPv6 protocol where backwards compatibility is questionable at best.

In an effort to test and develop IPv4/IPv6 convergence techniques, many of the common services expected of a network were used as a baseline for testing. Many of the services were, however, limited more by the use of common, mainstream operating systems such as Windows Server 2003. It has been documented, though not fully tested, that DNS BIND has been successful as an IPv6-only DNS. Other documentation has suggested the existence of NTP and SNMP services in an IPv6-only environment. These documents are important as they demonstrate the current viability for IPv6-only networks that provide the same services as common IPv4 networks. However, the current latest version of Windows Server limits the usability of many of the services to strictly an IPv4/IPv6 network. To be sure, while the eventual goal is that IPv6 replace IPv4, coexistence of the two protocols must be allowed as IPv4 will be around for long time yet. What the implementer must be cautious of is that applications, software, and operating systems do not become the limiting reagent for the process of change. As the research study has shown, Windows Server 2003 could be such a limiting reagent if relied upon.

The third goal was not entirely met. This was largely due to the difficulties in setting up an IPv6 network with common, up-to-date systems. However, the fact that the implementation of IPv6 was difficult on today's systems gives a hint at the possible difficulties in attempting to implement IPv6 on legacy systems. One area that shows promise, however, is the use and implementation of Linux and Unix-based operating systems. Many Linux and Unix based systems already have IPv6 enabled. Others have the capability, though it is disabled by default. Also, as Linux and Unix-based systems tend to be more easily updateable and patchable, the limit to these systems is more concentrated with the physical hardware than with the capability of the OS.

Ergo, in conclusion, IPv6 is a new protocol that provides many fundamental changes to the widely standardized and implemented IPv4 protocol. As shown previously, the global need for addresses and a better protocol demands the implementation of IPv6. However, the implementation of this new protocol is limited largely by the software and systems widely used and the alacrity with which major software vendors pursue the ability to seamlessly implement IPv6 in a largely IPv4 world.

Appendix A: Log

06SEP07:

- Installed IPv6 on computers running Windows XP Professional SP2. Used "netsh netsh interface ipv6 install".
- Connected the three computers to a four port hub.
- Set up computer (Windows XP) as router (IPv6).
    - netsh interface ipv6 set interface "Local Area Connection" forwarding=enabled advertise=enabled
- Successfully pinged each computer from each computer.
- Following Setup:
    1. PANDORA -> fe80::208:74ff:fe39:90d2
    2. DAEDALUS -> fe80::208:74ff:fe39:90d4
    3. ICARUS -> fe80::208:74ff:fe39:9105

08SEP07:

- Installed Wireshark on all computers (computers running Windows XP Professional SP2):
    - Use filter: "ipv6"
- Attempted to setup FTP server - need XP Professional install discs.
- Attempted to setup telnet:
    - Faced user/login issues
    - Logged in once, telnet crashed (may be error in setting up)
- tested Buffer size of ICMPv6: 1452

10SEP07:

- "You have to have the connection from the wall plugged into Fastethernet 0/1. Do not change this connection it will not work. Thanks - B.Lucas & D. Christenson"

16SEP07:

- Linux (SUSE) computer could not login (contact Becker).
- Telnet works between DAEDALUS (fe80::208:74ff:fe39:90d4) and ICARUS (fe80::208:74ff:fe39:9105)
    - Connection achieved (had to add Port 23 to Windows Firewall exceptions)
    - Netstat confirmed connection with IPv6 address listed
- FTP (File Transport Protocol)
    - Connection problem on port 21
    - Ran into issue when ftp host on port 23 (telnet port). After 60 seconds, connection closed by remote host.
    - Open port 2345 on both computer's firewall
        - Still ran into connection issue ("ftp: connect :unknown error number").
- Will load NetCat and NMap to further test connectivity between computers on IPv6 Network

---

18SEP07

- Cannot ping or connect computers (all running Windows XP Professional SP2). Found computers turned off. Error: "Destination unreachable".
- Problem solved. XP uses a local index at the end of an IPv6 address. Must check upon star-up with the following command: "netsh interface ipv6 show interface"
- All computers pingable. Telnet works as well.

24SEP07

- SUN Solaris not installed yet. Wit till then to set up IPv6 network on UNIX/LINUX machines.
- Attempted MAC spoofing to determine if it would change IPv6 link-local address.
    - Used T&R SoftNet Solutions MACSpoof to spoof MAC address.
    - IPv6 link-local address change successful
    - Implications: one can still spoof their IPv6 link-local IP address (or any address that uses their IPv6 Interface ID) by spoofing their MAC address.
- Installed IIS 6.0, enabled FTP
    - Attempted both Active as well as Passive FTP.
    - All attempts failed.
    - Message: ">ftp: connect :unknown error number"
    -
    - Modified and restarted Firewall settings
    - All attempts failed

26SEP07

- SUN Solaris installed on computers
- Loaded XLight FTP virtual server on PANDORA
    - Selected "Dynamic IPv6" port 21
    - User/Password: ********/*******
    - created remote admin account
        - Port 3333
        - User/Password: ********/*******
    - FTP Connection:
        - Connection successful
        - DIR -> list filed/directory: FAILED
            - FTP command line would state following and freeze: "150 Opening ASCII mode data connection for /bin/ls (x bytes)" where x = # of bytes
            - Attempted Active: FAILED - see above
            - Attempted Passive: FAILED - see above
            - Attempted Binary: FAILED - see above
            - Attempted ASCII: FAILED - see above
            - Attempted "cd testmaterial": SUCCESS
            - Attempted "get": FAILED
                - Would create local file
                - Failed to send/receive file data
            - Attempted Remote Administration: SUCCESS
                - Connection fully functional
                - Connected DAEDALUS to PANDORA successfully
    - Due to 3rd Party FTP Failure, Installed Windows Server 2003 Edition

---

27SEP07

- Set up File Server and File Transport Protocol server on Windows Server 2003, Enterprise Edition.
  - Attempted to ping: SUCCESS - was able to ping other computers
  - Attempted FTP: FAILURE - message: ">ftp: connect :unknown error number".
- Set up switches, connected Solaris computers to switch, connected all switches. All pings: SUCCESS!

30SEP07

- Goals:
  1. Ping USMA
  2. FTP Windows
  3. Set up CISCO 1811; if time, via linux
- Ping USMA
  - COM1, 9600Bits/sec; Databits 8, Parity 1; flow Control Hardware
  - Attempted to ping Army at following locations:
    - 134.240.18.81 -> FAILURE
    - 134.18.9.49 -> SUCCESS
    - 134.8.44.2 -> SUCCESS
  - Traceroute 134.240.18.81:
    - 131.122.204.254
    - 192.190.228.1
    - 10.10.1.51
    - 138.18.9.49
    - 138.18.44.2
- FTP Windows
  - Started XLight FTP Server
    - Connection from DAEDALUS: SUCCESS
    - Directory listing ("dir"): SUCCESS
    - Transfer of text file: SUCCESS
  - Normal Windows FTP
    - Connection from DAEDALUS: FAILURE
- Set-up CISCO 1811
  - Attempted to set up router: FAILURE. This believed to be in part due to either human error or an incompatibility with the CISCO 1811 and IPv6. While the IOS for the CISCO 1811 is IPv6 compatible, doubt exists as to its physical compatibility.

02OCT07

- Attempted to create DNS (Domain Name Service server) on PANDORA using Windows Server 2003, SP1.
  - Ensure IPv6 was installed and enabled
  - Set up network to following:
    - IP: 10.0.1.0

        - Subnet Mask: 255.255.255.0
    - Set PANDORA to 10.0.1.2
    - Set DAEDALUS to 10.0.1.3
    - Set ICARUS to 10.0.1.4
    - Set Gateway to 10.0.1.1
    - Enabled forwarding and advertise on each computer
    - Set up forward lookup xone: (testlab.ipv6.com)
    - Note: do not have AD set up (incompatible according to website)
    - Set DNS address (IPv6) to FEC0:0:0:FFFF::1%1
    - Set as DNS address for each client machine
    - Test:
        - Ping using IPv6: SUCCESS
        - Ping using IPv4: SUCCESS
        - Nslookup using IPv6: FAILURE
        - Nslookup using IPv4: FAILURE

03OCT07

- Create IPv6 Testlab with fully functioning DNS;
    - Re-installed Windows Server 2003 on PANDORA: successful
        - Set up as standalone server
        - Set up DNS server (installed networking services)
        - Defined Forward Lookup Zone (testlab.ipv6.com)
        - Set IP to 10.0.1.2/24
        - Installed IPv6 (netsh interface ipv6 install)
    - Re-installed Windows XP SP2 on ICARUS: FAILURE
        - Problem due to corrupt boot disk image – need to acquire other image
    - DAEDALUS
        - Uninstalled IPv6 (netsh interface ipv6 uninstall)
        - Installed IPv6 (netsh interface ipv6 install)
        - Set IP to 10.0.1.3/24 and DNS to 10.0.1.2
        - Appended NDS suffix “testlab.ipv6.com”
        - Configured Windows Firewall to allow incoming echo request.
        - Note: changed “Local Area Connection” name to “Network 1 Connection” on both]
        - Conducted ping of FE80::208:74FF:FE39:9102 : successful
        - Enabled forwarding and advertising
        - Published route as 2001:DB8:0:1::/64
    - Ipconfig showed global addresses for both.
    - Created record for DAEDALUS in PANDORA (DNS) [IPv6] using IPv6 global address
    - “ping DAEDALUS” : successful
    - “ping PANDORA” : successful
    - DNS LOOKUPS CONSIDERED SUCCESSFUL

04OCT07

- Set up DNS on all computers
    - Finished installing Windows XP PRO SPD2 on ICARUS
        - Installed IPv6
        - Set IPv4 IP to 10.0.1.4/24

        - Appended DNS suffix to “testlab.ipv6.com”
        - Ping DAEDALUS: successful
        - Ping PANDORA: successful
    - Set up DNS zone on Solaris computers:
        - Modified /etc/nsswitch.conf
            - It was modified for the IPv4 and IPv6 address being used in the setup.
        - Created /etc/resolv.conf:
            - search testlab.ipv6.com
            - domain testlab.ipv6.com
            - nameserver 10.0.1.2
        - Modified /etc/inet/netmasks
            - Changed IPv4 address scheme to 10.0.1.0 255.255.255.0
        - ifconfig elxl0 10.0.1.x netmask 255.255.255.0 up [where x denotes what machine this is being configured on]
    - Successfully pinged each computer

08OCT07

- Modified /etc/ftpd/ftpusers to alow connection by user “root”.
- FTP attempts:
    - All successful.

10OCT07

- Pinged USMA
    - Ping 46% successful
    - IPv6 Address: 2001:1918:F100:1::1
    - Reason for previous difficulty: usma router st up to drop ICMPv4 packets (Access-Control Lists)
    - Ping success rate not optimal
- Set up Web Server
    - Set up PANDORA as an Application Server
    - Installed ASP.NET and IIS 6.0
    - Created Webpage (“index.htm”)
    - Access Webpage
        - Successful if IPv6 IP is put in brackets: “http://[2001:DB8:0:1:208:74FF:FE39:90D2]/index.htm”
        - Note: Needs authentication to view.
        - Can bypass using DNS lookup names:
            - “http://PANDORA/index.htm”
        - Need to figure out permissions
    - Note: IE v6.0 is NOT IPv6 compatible. Mozilla Firefox is.

15OCT07

- Connected USNA 3600 to 8 port switch (0/0)
- Deleted route 2001:DB8:0:1::/64
- 3600 “assigned” 2001:1918:F101::/48

- Tested ping of 2001:1918:F101:1::1
  - Success.
- Tested ping of 2001:1918:F100:1::1
  - Success.
- Tested ping of 2001:1918:F100:1::1 on DAEDALUS
  - Success.
- Tracert of 2001:1918:F100:1::1 on DAEDALUS
  - Success
- Assigned 2001:1918:F100:1::1 to DNS as USMA

23OCT07

- Installed dnscmd.exe onto PANDORA
- Entered IPv6 'AAAA' Records for all computers
- Attempted to setup DNS as strictly IPv6
  - Failed: this is most likely due to static routing issues
  - Need to create static route for cisco 3600
- Re-Setup 6over4 network for DNS
  - Success
- Tested Website
  - Success
- Tested nslookup (reverse zone lookups)
  - Success

06NOV07

- Attempted to set up NTP on the network
  - Edited registry of PANDORA to use internal clock
  - Attempted to connect to PANDORA as NTP server with ICARUS
    - Failed: could not reach host.
- Most likely due to Windows Time Service executing as a group policy.

Appendix B: Router Configuration File

```
no ip dhcp use vrf connected

usna-3660#show conf
Using 2190 out of 129016 bytes

version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption

hostname usna-3660

boot-start-marker
boot-end-marker

enable password usna

no aaa new-model

resource policy

ip subnet-zero

ip cef
no ip domain lookup
no ip dhcp use vrf connected

ipv6 unicast-routing
ipv6 host usma FEC0:1918:F100::1
ipv6 host usna-2600 FEC0:1918:F101::200
ipv6 host usma-2600 FEC0:1918:F100::2
ipv6 host usna FEC0:1918:F101::1
ipv6 cef
ipv6 multicast-routing

username cisco password 0 cisco

interface Tunnel62
 description usna-usma tunnel-type 41
 no ip address
 ipv6 enable
 ipv6 ospf 27133 area 0
 tunnel source FastEthernet0/1
 tunnel destination 134.240.18.81
 tunnel mode ipv6ip
```

```
interface Tunnel63
 description usna-usma tunnel-type 47 GRE
 no ip address
 ipv6 enable
 ipv6 ospf 27133 area 0
 tunnel source FastEthernet0/1
tunnel destination 134.240.18.81

interface Loopback6
 no ip address
 ipv6 address FEC0:1918:F101::1/128
 ipv6 ospf 27133 area 0

interface FastEthernet0/0
 no ip address
 duplex auto
 speed auto
 ipv6 address 2001:1918:F101:1::1/64
 ipv6 enable
 ipv6 ospf 27133 area 0

interface FastEthernet0/1
 ip address 131.122.204.1 255.255.255.0
 duplex auto
 speed auto

interface ATM2/0
 no ip address
 shutdown
no atm ilmi-keepalive

interface Serial3/0
 no ip address
 shutdown
 serial restart-delay 0

interface Serial3/1
 no ip address
 shutdown
 serial restart-delay 0

interface Serial3/2
 no ip address
 shutdown
 serial restart-delay 0
```

```
interface Serial3/3
 no ip address
 shutdown
 serial restart-delay 0

interface FastEthernet4/0
no ip address
 half-duplex

interface FastEthernet5/0
 no ip address
 shutdown
 duplex auto
 speed auto

interface FastEthernet5/1
 no ip address
 shutdown
 duplex auto
 speed auto

router ospf 27133
 log-adjacency-changes

no ip http server
no ip http secure-server
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/1

ipv6 router ospf 27133
 router-id 131.122.204.1
 log-adjacency-changes

control-plane

line con 0
login local
line aux 0
line vty 0 4
login local

end

usna-3660#
```

Appendix C: Lab Setup (Visual)

Figure C-1

Figure C-2

References

[1] "Department of Defense Internet Protocol Version 6 Generic Test Plan Version 2", Defense Information Systems Agency, 2006; http://jitc.fhu.disa.mil/adv_ip/register/docs/ipv6_gtp_v2.pdf

[2] U.S. Service Academy IPv6 Pilot Network Project, DISA Project Proposal, March, 2007.

[3] P. Loshin, *IPv6: Theory, Protocol, and Practice*, 2nd ed., Morgan Kaufmann Publishers, 2004.

[4] C. Popoviciu, E. Levy-Abegnoli, and P. Grossetete, *Deploying IPv6 Networks*, Cisco Press, 2006.

[5] R. Hinden, *Internet Protocol Version 6 (IPv6) Addressing Architecture*, IETF RFC 3513, 2003; http://www.faqs.org/rfcs/rfc3513.html.

[6] R. Hinden, *IPv6 Global Unicast Address Format*, IETF RFC 3587, 2003; http://www.faqs.org/rfcs/rfc3587.html.

[7] S. Deering, *IPv6 Scoped Address Architecture,* IETF RFC 4007, 2005; http://www.faqs.org/rfcs/rfc4007.html.

[8] T. Berners-Lee, *Uniform Resource Identifiers (URI): General Syntax*, IETF RFC 2396, 1998; http://www.faqs.org/rfcs/rfc2396.html.

[9] IAB, *IAB/IESG Recommendations on IPv6 Address Allocations to Sites*, IETF RFC 3177, 2001; http://www.faqs.org/rfcs/rfc3177.html.

[10] Microsoft Corporation, "Step-by-Step Guide for Setting Up IPv6 in a Test Lab", Microsoft Corporation, 2006.

[11] C.B. Landis, "IPv6 Testing", Computer Science Department Research Project Report, Technical Report 2006-02, US Naval Academy, 2006.